APPLICATION FOR
UNITED STATES PATENT
IN THE NAME

Of

JOSEPH C. STURTEVANT AND SUHRUD I. DAGLI

FOR

SYSTEM AND METHOD FOR AGGREGATING DATA HAVING DATA AND APPLICATION DEPENDENCIES

Attorney Docket No. 51456.00003

Please direct communications to:

Squire, Sanders & Dempsey L.L.P.
600 Hansen Way
Palo Alto, CA 94304-1043
(650)856-6500

Express Mail Number: EL 701 316 102 US

SYSTEM AND METHOD FOR AGGREGATING DATA HAVING DATA AND APPLICATION DEPENDENCIES

PRIORITY REFERENCE TO PRIOR APPLICATIONS

5   This application claims benefit of and incorporates by reference

patent application serial number 60/256,133, entitled "Newbreak

Platform-Independent Aggregation Enabling Architecture With

Application And Data Dependencies For A Custom Functional Flow," filed

on December 13, 2000, by inventors Suhrud I. Dagli and Joseph C.

10   Sturtevant.

Technical Field

This invention relates generally to aggregating data, and more

particularly, but not exclusively, provides a system and method for

15   aggregating data having data and application dependencies.

Background

Conventionally, to view data from a variety of sources, such as

websites, local applications, external applications, trading exchanges,

20   etc., a user must access a plurality of interfaces, which can be

complicated and time consuming. Further, to generate a customized

solution, data from one interface may need to be integrated into a second

interface, often via manual intervention, which can be very inconvenient.

Accordingly, a new system and method for aggregating data with

25   application and data dependencies may be needed.

SUMMARY

The present invention provides a system for aggregating data with application and data dependencies. The system comprises an aggregator system communicatively coupled to a client system. The aggregator

5   system can also be communicatively coupled to external data and application sources, such as web services, web sites, trading exchange feeds, market data feeds from external sources, external databases, specific applications and spreadsheets from computers within a client's local network.

10   The client system comprises a client browser, an aggregation client, and a screen manager client. The client system may also comprise at least one local application and local data. The aggregation client communicates with the screen manager client and local and external sources for integrating data and applications. Specifically, the

15   aggregation client receives subscription requests from the screen manager client; creates a persistent HTTP connection with the aggregator system for all independent subscriptions; maintains the connection with the aggregator system and reestablishes the connection if it is terminated; communicates new subscription requests to the aggregator

20   system; creates subscription links for all local services; and communicate with the screen manager client to update application windows on receipt of updates from subscriptions. In addition, the aggregation client further maintains a map of all dependent

subscriptions to applications to enable updating of data when dependent data changes.

The aggregator system comprises a server engine and a request processor engine. The server engine is capable to aggregate data from a plurality of external sources, interfaces and program output. The server engine forwards aggregated data to the client system upon receipt of a command from the client system via the request processor engine.

The present invention further provides a method for aggregating data with application and data dependencies. The method comprises selecting or creating an application; registering the application with screen manager client; registering the application with the aggregation client; registering with appropriate sources to receive data for the registered application; receiving data from the sources; determining if any of the registered applications is dependent on the received data or dependent on data generated by an application based on the received data; if an application is dependent, forwarding the required data (either received or generated by another application) to the application for processing; and displaying updated data comprising the received data and/or generated data and/or processed data from a dependent application.

The system and method may advantageously enable the aggregation of data having data and/or application dependencies.

BRIEF DESCRIPTION OF THE DRAWINGS

Non-limiting and non-exhaustive embodiments of the present

invention are described with reference to the following figures, wherein

like reference numerals refer to like parts throughout the various views

5   unless otherwise specified.

FIG. 1 is a block diagram illustrating a network system for

aggregating data having application and/or data dependencies in

accordance with an embodiment of the invention;

FIG. 2 is a block diagram illustrating an example computer system;

10   FIG. 3 is a block diagram illustrating the aggregator system of the

aggregator server of FIG. 1;

FIG. 4 is a block diagram illustrating the client system of the client

of FIG. 1;

FIG. 5A and FIG. 5B are flowcharts illustrating a method for

15   aggregating data having data and/or application dependencies; and

FIG. 6 is a diagram illustrating an example screen produced by a

screen manager client of the client system of FIG. 4.

DETAILED DESCRIPTION OF THE ILLUSTRATED EMBODIMENTS

The following description is provided to enable any person skilled in the art to make and use the invention, and is provided in the context of a particular application and its requirements. Various modifications

5 to the embodiments will be readily apparent to those skilled in the art, and the principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the invention. Thus, the present invention is not intended to be limited to the embodiments shown, but is to be accorded the widest scope

10 consistent with the principles, features and teachings disclosed herein.

FIG. 1 is a block diagram illustrating a network system 100 for aggregating data having application and/or data dependencies in accordance with an embodiment of the invention. System 100 comprises an aggregator server 130, a client 145 communicatively coupled to the

15 server 130, and an external source 110 communicatively coupled to the server 130. For security reasons, system 100 may also include a first firewall 125 between the server 130 and the external source 110 and a second firewall 140 between server 130 and client 145. In another embodiment of the invention, additional servers or other sources of data

20 and/or applications may be communicatively coupled to the server 130.

In another embodiment of the invention, network system 100 may further comprise a mobile device (not shown), such as a laptop or handheld computer, communicatively coupled to the server 130. The

mobile device may also include a client system so as to enable aggregation from multiple sources that may have data and/or application dependencies.

In another embodiment of the invention, network system 100 may
5 further comprise one or more external clients (not shown) communicatively coupled to server 130. The external client may comprise a client system similar to client system 150 and can update data stored in client system 150.

Aggregator server 130 comprises an aggregator system 135 for
10 aggregating data and applications from external source 110 and/or other sources (not shown) for transmission to client 145. System 135 will be discussed in further detail in conjunction with FIG. 3. Client 145 comprises client system 150 for aggregating data and applications from system 135 and for presenting the data in application windows to a user
15 for viewing. System 150 will be discussed in further detail in conjunction with FIG. 4. External source 110 may comprises an external application 115 for processing data and external data 120. External source 110 may include a proprietary model, trading exchange or any other type and source of applications and/or data.

20 FIG. 2 is a block diagram illustrating an example computer in accordance with the present invention. In an embodiment of the invention, client 145, aggregator server 130, and external source 110 may include or be resident on example computer 200. The example

computer 200 includes a central processing unit (CPU) 205; working memory 210; persistent memory 220; input/output (I/O) interface 230; display 240 and input device 250, all communicatively coupled to each other via system bus 260. CPU 205 may include an Intel Pentium® 5 microprocessor, a Motorola Power PC® microprocessor, or any other processor capable to execute software stored in persistent memory 220. Working memory 210 may include random access memory (RAM) or any other type of read/write memory devices or combination of memory devices. Persistent memory 220 may include a hard drive, read only 10 memory (ROM) or any other type of memory device or combination of memory devices that can retain data after example computer 200 is shut off. I/O interface 230 is communicatively coupled, via wired or wireless techniques, to a network for communicating with other computers. In an alternative embodiment of the invention, I/O 230 may be directly 15 communicatively coupled to a server or computer. Display 240 may include a cathode ray tube display or other display device. Input device 250 may include a keyboard, mouse, or other device for inputting data, or a combination of devices for inputting data.

One skilled in the art will recognize that the example computer 200 20 may also include additional devices, such as network connections, additional memory, additional processors, LANs, input/output lines for transferring information across a hardware channel, the Internet or an intranet, etc. One skilled in the art will also recognize that the programs

and data may be received by and stored in the computer 200 in alternative ways.

FIG. 3 is a block diagram illustrating the aggregator system 135 of the aggregator server 130 (FIG. 1). System 135 comprises a server

5 engine 300, a request processor engine 310, an application 320 and data 330. System 135 components, such as engine 310, may be implemented with software, integrated circuits, digital signal processors and/or other devices. In an embodiment of the invention, the system 135 may further comprise additional or fewer applications and/or no data 330. Server

10 engine 300 maintains a persistent HTTP connection (port 80) with the client system 150 and receives data from external sources such as external source 110 and from local sources such as application 320 and data 330. External sources can include proprietary models, back office, legacy systems, spreadsheets, vendor models, trading exchanges, the

15 Internet, manual input, and other sources. Via the persistent connection, the server engine 300 can send the received data to the client system 150. All data sent to client system 150 may be encrypted to enable high security use. Engine 310 receives requests for data from client system 150 and instructs server engine 300 to transmit the

20 requested data to client system 150. Application 320 resides in aggregation system 135 and generates data for display on client 145. Examples of functions of application 320 include mortgage rate calculations, news aggregation, bond price calculations, etc.

FIG. 4 is a block diagram illustrating the client system 150 of the client 145 (FIG. 1). Client system 150 comprises a client browser 400, an aggregation client 410, a screen manager client 420. In addition, client system 150 may further comprise a local application 430 and local

5 data 440. System 150 components, such as aggregation client 410, may be implemented with software, integrated circuits, digital signal processors and/or other devices. Client browser 400 may include an Internet web browser such as Internet Explorer or Netscape Navigator®. Browser 400 can display data generated by applications and/or from

10 subscriptions in application windows. Application windows will be discussed in further detail in conjunction with FIG. 6.

Aggregation client 410 communicates with screen manager client 420 and local and external services or subscriptions for data aggregation. Client 410 receives application subscription requests from the screen

15 manager client 420; creates a persistent HTTP connection with the server 130; maintains the connection and reinitiates the connect in case of severance; communicates with the request processor engine 310 to inform the server engine 300 of any new application subscription requests or changes in existing subscriptions; creates application

20 subscriptions links with local applications; communicates with the screen manager client 420 to update application windows displaying data upon receiving updates from subscriptions; maintains a map of all

dependent applications; and receives updates for all independently updating applications from a local source and/or from server 130.

Screen manager client 420 manages application windows on browser 400. Specifically, the client 420 can create application windows;

5   subscribe to update procedures for applications via the aggregation client 410; interface with the aggregation client 410 for manual updates; apply updates to application windows; maintain a list of all application windows, their corresponding application subscriptions and dependencies; maintain layout information for the application windows;

10   provide screen management functions to freeze layout and lock window positions. Further, in an embodiment of the invention, a user can change layout and window positions.

Local application 430 may generate data based on received data or local data from a subscription, may display received data from a source

15   without modification or may generate data based on data generated from a second application (not shown) for display via browser 400. For example, application 430 generated data may be dependent on received data or data generated by another application. Local data 440 may include data generated by application 430 and/or may be any other data

20   stored locally.

FIG. 5A and 5B are flowcharts illustrating a methods 500a and 500b for aggregating data having data and/or application dependencies. In an embodiment of the invention, multiple instances of methods 500a

and/or 500b may run simultaneously. In method 500a, an application

is first selected or created (510) and then run. The application can be a

local application, such as application 430, an application on server 130,

such as application 320, or an application located on an external source,

5   such as application 115 on external source 110. If the selected

application is not available locally (e.g., not local application 430), the

selected application can be downloaded to client 145 and then run.

The selected or created application is then registered (520) with the

screen manager client 420 so that client 420, in conjunction with

10   browser 400, can open (525) an application window for displaying

application output data.

Next, the application is registered (530) with the aggregation client

410, which then maps (540) the registered application's dependencies, if

any, to other applications or data. Sources required by the registered

15   application are then registered (550) with so that the sources can provide

data to applications on client system 150. Sources of data may include

JAVA applications, JAVA servlets, Excel spreadsheets, Corba services,

TCP/IP, DLLs, databases, legacy applications, text/ASCII files, XML

documents, other applications, etc. The sources may be local, such as

20   local data 440, or external, such as data 330 on server 130 or external

data 120 on external source 110. The method then ends.

In method 500b, updates are requested (560) from sources, either

from local sources or other sources via aggregator system 135 by sending

a request to request processor engine 310. Other sources may include

aggregator system 135 data and applications, external data and

applications, and user exported applications and data. Request for

updates may be initiated for different applications at different rates. In

5   addition, requests for updates may be initiated at regularly scheduled

intervals or at intervals specified by the applications.

Next, data is received (565) for independent applications, i.e.,

applications having no dependencies. The applications then are updated

(570) with the received data and application windows corresponding to

10   the applications are updated. It is then determined (575) if any

applications not updated are dependent on the updated applications.

The aggregation client 410 can make this determination using the

dependency map previously generated (540, FIG. 5A). If no applications

are dependent, the method 500b ends. Otherwise, the dependent

15   applications are updated (580) using data generated or received by the

independent applications. Screen manager client 420 can then display

output from the dependent applications in corresponding application

windows. The method 500b then ends.

FIG. 6 is a diagram illustrating an example screen 600 produced

20   by screen manager client 420 of the client system 150 (FIG. 4). Screen

600 comprises six application windows corresponding to six applications.

Application window 610 list bond prices from data feeds hosted on

aggregator server 130. Application window 620 lists mortgage rates

generated by an Excel spreadsheet. The mortgage rates are dependent on bond prices from application window 610. Accordingly, mortgage rates in window 620 are not updated until bond prices in window 610 are updated.

5   Application window 630 contains a chart generated by an application on aggregator server 130 using data obtained from a local database and server 130 models. Application window 640 lists yields for derivative bonds generated by an application on server 130. The yields are dependent on bond prices from application window 610 and a local

10   DLL stored on client 145. Accordingly, the yields are not updated until updated data is received for application window 610. Application windows 650 and 660 include news from news feeds hosted on server 130. The application(s) that generated windows 650 and 660 use data from application window 610 as a search term. Accordingly, the

15   application corresponding to windows 650 and 660 is dependent on data from the application that generates window 610.

The foregoing description of the preferred embodiments of the present invention is by way of example only, and other variations and modifications of the above-described embodiments and methods are

20   possible in light of the foregoing teaching. Although the network sites are being described as separate and distinct sites, one skilled in the art will recognize that these sites may be a part of an integral site, may each include portions of multiple sites, or may include combinations of single

and multiple sites. Further, components of this invention may be implemented using a programmed general purpose digital computer, using application specific integrated circuits, or using a network of interconnected conventional components and circuits. Connections may

5   be wired, wireless, modem, etc. The embodiments described herein are not intended to be exhaustive or limiting. The present invention is limited only by the following claims.